Naval Research Laboratory
Washington, DC 20375-5320

NRL/MR/6040--14-9480

CT-Analyst GIS Data Processing Guidance Document

Adam J. Moses
Laboratories for Computational Physics and Fluid Dynamics

April 30, 2014

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)* 30-04-2014	**2. REPORT TYPE** Memorandum Report	**3. DATES COVERED** *(From - To)*
4. TITLE AND SUBTITLE CT-Analyst GIS Data Processing Guidance Document		**5a. CONTRACT NUMBER**
		5b. GRANT NUMBER
		5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Adam J. Moses		**5d. PROJECT NUMBER**
		5e. TASK NUMBER
		5f. WORK UNIT NUMBER 64-9111-L-2
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Research Laboratory 4555 Overlook Avenue, SW Washington, DC 20375-5320		**8. PERFORMING ORGANIZATION REPORT NUMBER** NRL/MR/6040--14-9480
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Naval Research Laboratory 4555 Overlook Avenue, SW Washington, DC 20375-5320		**10. SPONSOR / MONITOR'S ACRONYM(S)** NRL
		11. SPONSOR / MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

CT-Analyst is a hazardous plume modeling application developed by the Naval Research Laboratory in Washington, DC. CT-Analyst adapts to limited known source information by pre-calculating from modeling runs done on a variety of modeling conditions and differing prevailing wind directions and velocities. This pre-calculation is done by way of highly complex, highly-accurate Computational Fluid Dynamic (CFD) computer codes run on high-performance computer systems, which produce usable Nomograf tables. While these Nomograf tables serve as the output from these CFD codes, the input provided are essentially 3-dimensional models of the region or area-of-interest that is intended to be modeled, like a city, military base, industrial plant, etc., and fully describe the ground terrain, buildings, trees, and water. The models require specific input datasets, whose construction and development should be described in detail.

15. SUBJECT TERMS

Computational fluid dynamics	CT-Analyst	Visualization
Geographic information systems	Modeling	

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Adam J. Moses
a. REPORT Unclassified	**b. ABSTRACT** Unclassified	**c. THIS PAGE** Unclassified	Unclassified Unlimited	19	**19b. TELEPHONE NUMBER** *(include area code)* (202) 767-2556

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

Contents

Introduction

CT-Analyst is a hazardous plume modeling application developed by the Naval Research Lab in Washington, DC. CT-Analyst provides high-fidelity, highly-accurate contaminant plume hazard information in very short order, nearly instantaneous. These hazard predictions provide not only coverage area from a hazard beginning at a given source location, but also concentration and health effect information as well. This data can then help facilitate first responders, military, and law enforcement in responding to a given event, such as a terrorist attack or industrial accident.

CT-Analyst achieves this rapid speed advantage in predicting these hazard plumes by way of pre-calculation. Many other similar modeling tools would wait until all the actual conditions are known before starting a modeling run, which then may take anywhere from ten minutes to an hour. CT-Analyst adapts to limited known source information by pre-calculating from modeling runs done on a variety of modeling conditions and differing prevailing wind directions and velocities.

This pre-calculation is done by way of highly complex, highly-accurate Computational Fluid Dynamic (CFD) computer codes run on high-performance computer systems. Once run these CFD codes produce Nomorgaf tables, which are a database of wind-field information that powers CT-Analyst.

While these Nomograf tables serve as the output from these CFD codes the input provided are essentially 3-dimensional models of the region or area-of-interest that is intended to be modeled, like a city, military base, industrial plant, etc, and fully describe the ground terrain, buildings, trees, and water. ***The specifications and construction of these input 3D models is what this document will hope to describe.***

Also note that while this document will hope to describe as much of this process as possible it can in no way describe all of it. In some cases the best way to get through tough problems when processing the data can is found in experimenting with it or really just doing something as simple as a Google search.

If any questions or comments or further help is required please contact the point of contact mentioned on the cover sheet of this document.

Manuscript approved May 23, 2013.

Required Inputs Data For Nomograf Computation

This section will describe the very specific formats required of the input data to the CFD codes that produce the Nomografs. Later sections will describe how to arrive at these formats but it is important to first know what you'll be looking to arrive at. After the descriptions there will be images included showing what is being described.

Format Specification:
Data input to the CFD codes are constructed as a number of binary files. There files are essentially just a gridded set of 2-dimeinstal arrays where each cell contains a height value, therefore when read in total it is possible to construct a 3-dimensional model of the area being modeled.

Below is a list of the input files, with a short explanation to follow. Some are strictly required, others are not. A more detailed description follows later.

- Ground Terrain Heights (required type)
- Building Heights (required type)
- Tree Heights (or at least locations)
- Water Heights (or at least locations)
- Additional Land Use Locations (i.e. roads, highways, marked areas of interest, etc)

Each of the file types listed above are all constructed the same.

- The files are raw binary, containing no headers.
- The files may be output in either big or little endian, although note the type for inclusion in the information file.
- Then files should be written in 2-byte unsigned integers.
- For height-value data that was expressed in floating point value meters, this should be first multiplied by twenty and the truncated to form the required 2-byte unsigned integer.
- If expressing non-height value data, i.e. in the land use location file, fixed small number values should be used. For instance, in a Land Use file, the value of 10 could equal roads, 20 could equal highways, etc.
- The values should be written in a table (raster) format, meaning rows and then columns. This means that the first 2-byte unsigned value encountered will be position (0, 0), the second value will be (1, 0), etc. This proceeds until the end of the first row, then immediately followed by the value for the second row, i.e. (2, 0).
- The table should start at the North-West corner of the region. This means position (0, 0) as mentioned should correspond to the most north-western point in the region, and the final position should be the most south-eastern.
- All files in a given set for a region should be identical in size, and should have the same number of rows and columns. Correspondingly, since they reflect an actual region, they should have corresponding dimensions and resolutions, as well as being geo-referenced to the same space.

Finally a set of "helper" files are also typically included, however these are not expected in the binary format mentioned above. A more detailed description follows later.

- Information File – just a regular text or Word file.
- Reference Image Of The Region – an image file, like a JPEG or BITMAP.

Required Types:
Ground Terrain Heights – This file should describe the bare-earth for the area being turned into a Nomograf. This is to mean that each value in the file should represent the height, in meters, at that location. This value can be either above sea-level (ASL), which is preferred, or simply relative to itself. In the case of being relative to itself this means that the lowest point should be made to be a value of 1, all other points would therefore then be higher than this, there cannot be negative values. Since this is a reflection of the ground there cannot be areas that contain no-values, since a region wouldn't be expected to have magical-void areas. This file is considered required, however if this file is not provided the model can be run where it will be assumed all the terrain is flat and the same height. (See Figure 1)

Building Heights - This file describes the buildings in the area being turned into a Nomograf. This is to mean that each value in the file should represent the height, in meters, at the location of everywhere there is a building. This value can be either above sea-level (ASL) or above ground-level (AGL). In places there are no buildings it is considered void, and therefore the value should be set to zero. Where there are buildings it is common that every value for a specific individual building the value will be the same everywhere, although this does not always have to be. If the building has a sloped roof or is domed for instance, and these values were reflected from the raw data, then a variety of values for the one building would occur. This file is required. (See Figure 2)

Extra Types:
Tree Heights – This file describes the trees in the area being turned into a Nomograf. This is to mean that each value in the file should represent the height, in meters, at every location there is a tree. Like buildings, this value can be either above sea-level (ASL) or above ground-level (AGL). In place there are no trees it is considered to be void and should be set to zero. In the event that only the location of trees is known but not individual heights, a default height of some kind may be used for all the trees, a good value to use here might be 5 meters, although this could change depending on the makeup of the area and the kind of trees found there. Unlike the composition of the buildings where much of the data may be expected to be squared and block-like looking, the tree data is often scattered, as are often the actual trees. If this file is not provided it will be assumed that no trees exist. (See Figure 3)

Water Heights – This file describes the water in the area being turned into a Nomograf. This is to mean that each value in the file should represent the height, in meters, at every location there is water. Like buildings, this value can be either above sea-level (ASL) or above ground-level (AGL). In places there are no water it is considered to be void and should be set to zero. In the event that only the location of water is known but not individual heights a value of 1 should be used. If this file is not provided it will be assumed that no water exists. (See Figure 4)

Additional Land Use Locations – This file describes other things of note in the area being turned into a Nomograf. While not used by any part of the actual computation, the land use is useful for later productions of the mask, and can be used to make more descriptive overlays. Instead of heights, simply identification values should be used, and can be picked arbitrary so long as these values are described elsewhere so it is known what they represent. For instance the file might contain the value 15 everywhere there is a road, 20 everywhere there is a highway, 30 everywhere there is a dock, etc. This file is optional, it is not required and will not change the model whether it is present or not. (See Figure 5)

Reference Material:
Reference Image Of The Region – An image, in a common image format like BITMAP of JPEG,

should also be included. This image can be derived as a composition of the overlays from the input data, or from another source like ArcGIS or Google Earth. It is used to get a sense of what the region looks like and that the modeling data output matches what is to be expected. (See Figure 6)

Information File – This file is key to defining the configuration of the computation that will produce the Nomografs. This file should be a basic text file that contains, at a minimum: number of rows of each of the tables; the number of columns of each table; the meter-resolution for each of cells, both for the x direction and y direction; the geo-referenced coordinates for each of the four corners of the area in the UTM reference system; the endian order of the files. Also if a Land Use file was included it should specify what value equal what, like 15 for a road, 20 for a highway, etc. (See Figure 7)

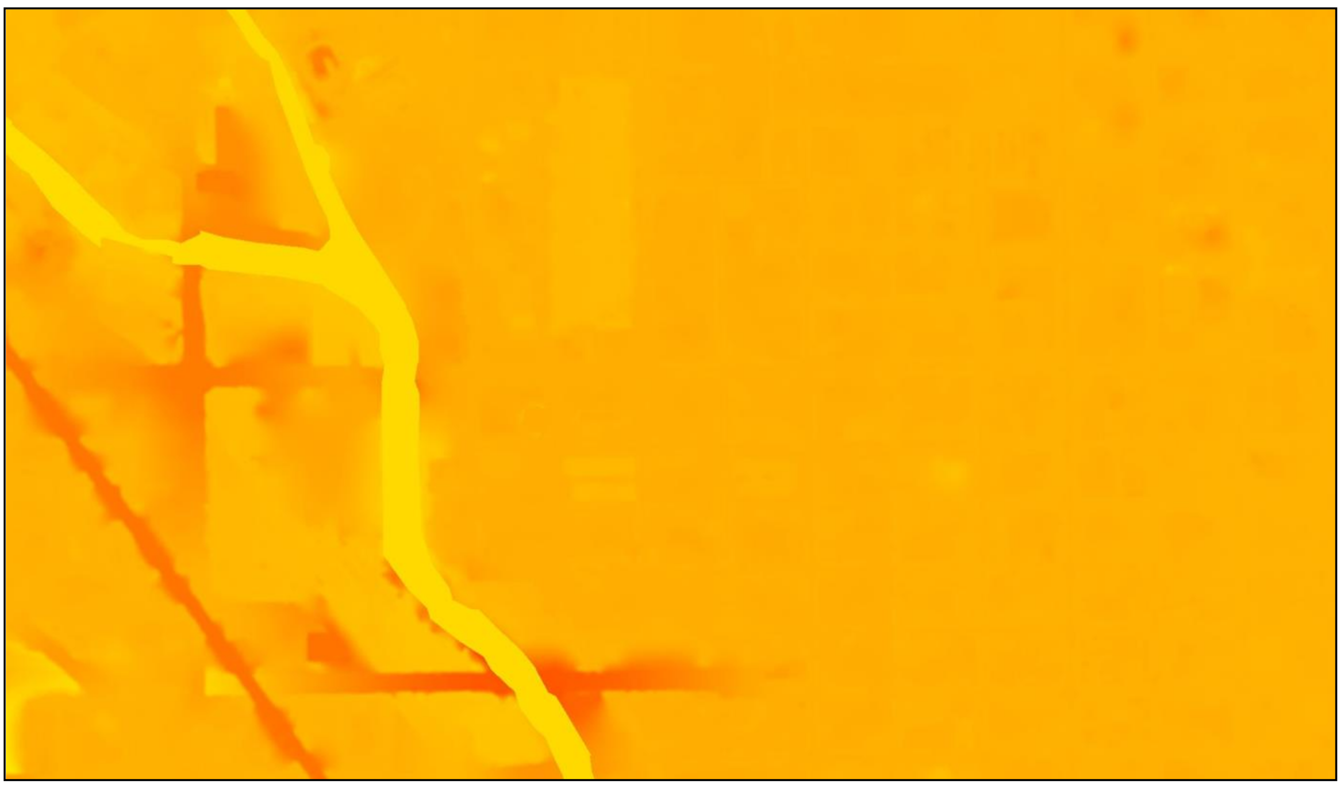

Figure 1. An example visualization of a ground terrain height file. Note that the colors shown are arbitrary; they are just picked and then scaled to whatever actual height values are contained in the data.

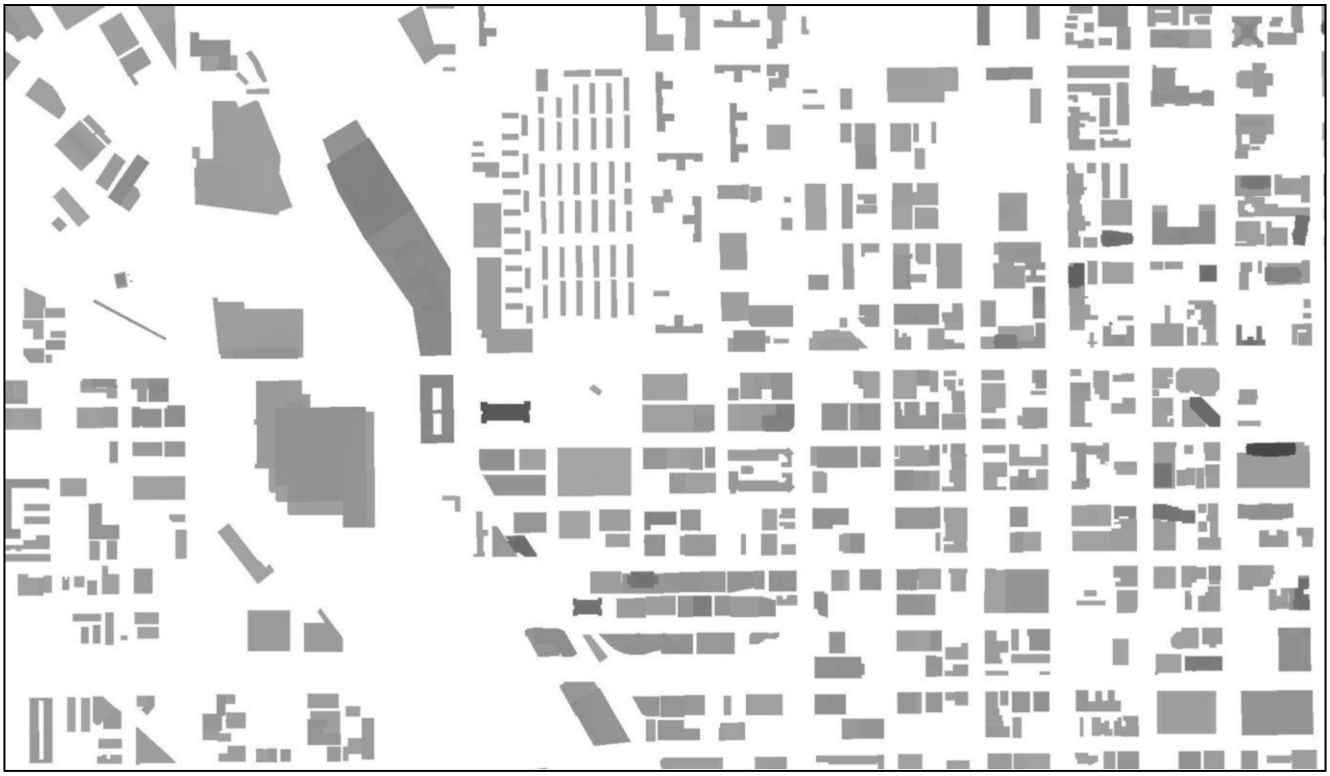

Figure 2. An example visualization of a buildings height file. Colors shown are arbitrary and based on heights in the data. The white areas are considered void and have no data, which in the binary file is expressed as zero.

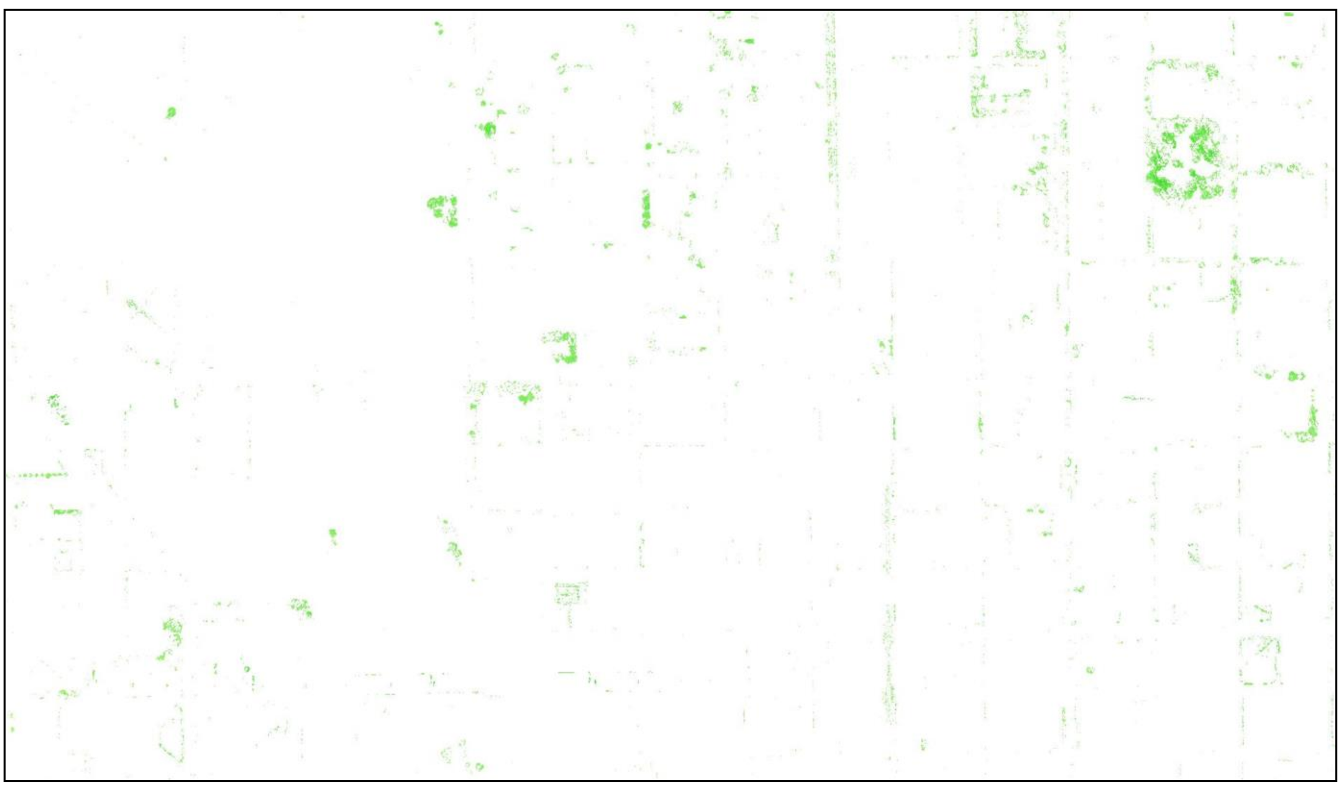

Figure 3. An example of tree heights file. Note most of the area is empty as few trees are present much in the region. Colors are also arbitrary here and the white area is encoded as zero.

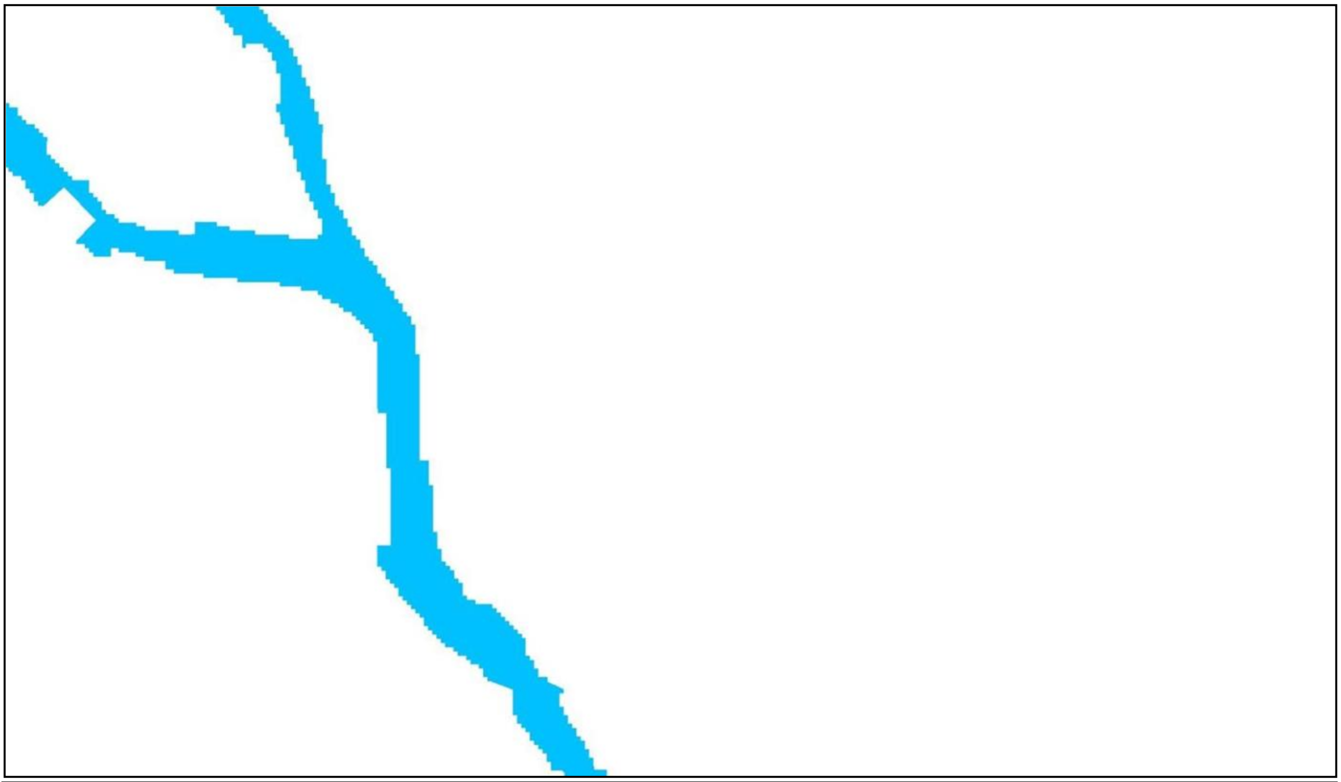

Figure 4. An example visualization of a water heights file. The colors are also arbitary here, there could be differences in height but they are shown all as solid blue.

Figure 5. An example visualization of a land use file, in this case showing roads which would be encoded to a specific value and the blank area encoded as zero. The color here is arbitrary.

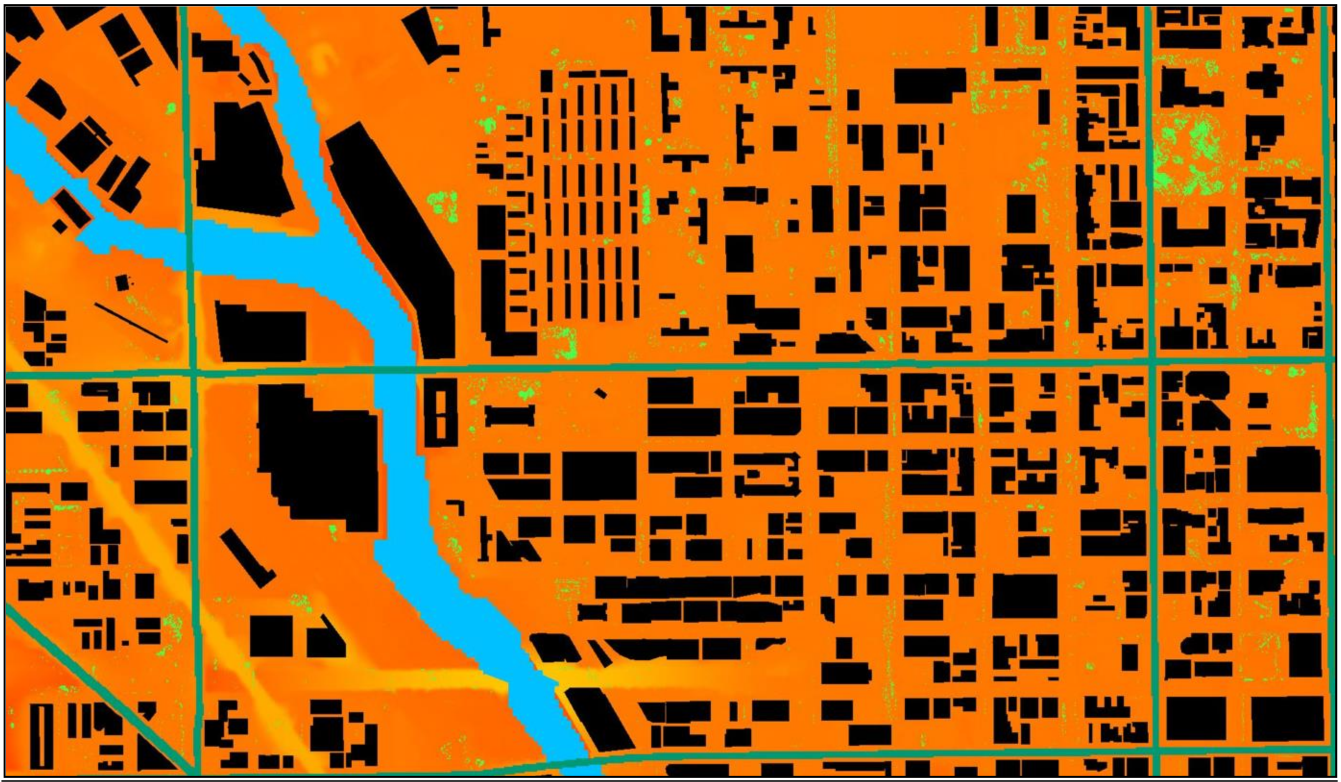

Figure 6. A reference image that is provided in addition to the binary files. It is essentially a composite of the earlier visualizations helping to identify the region being modeled.

```
16 byte fixed / Big-endian

4700 Cols (X)
8100 Rows (Y)

1 Meter Cell Resolution (X and Y)

Bounds
UTM 16 North
Upper Left  445500e 4641100n
Lower Right 450200e 4633000n

Land Use Values
10 is roads
```

Figure 7. An information file this is provided in addition to the binary files. It describes those binary files and the bounds of region being modeled. This file can be as simple as text entry in a text file.

Gathering GIS Data

Explanation:
In order to build the required input data sets described in the previous section you will first have to gather all the GIS data needed to do so. This starts off first by identifying the region you intend to model, Chicago for example. Once a region has been picked you'll also need to know where specifically would like to center the mode, downtown let's say, and then how far out from there you would like to go and in what directions. This provides your bounding box and helps you in searching for the required data you'll need.

This bounding box should be described specifically in UTM coordinates, denoting a north-west and south-east corner, which also provides an exact dimension in east-to-west (X) and north-to-south (Y) space. This part is important, so while you may chose to focus on downtown Chicago and scale back to say 5 kilometers wide and 8 kilometers tall, you need to settle on exact specific requirements for all of this. Using the example from Figure 7 you could choose precisely 4700 meters wide and 8100 meters tall with the exact outer boundary coordinates of 445500 easting and 4641100 northing to 450200 easting and 4633000 northing, using UTM Zone 16 North. If these numbers and coordinate types are unfamiliar to you they are done using UTM Coordinate System, which will be discussed in more depth in the following section. You will also want to be sure to pick boundaries that are divisible by 2 in both the X and Y directions, although usually picking something even more rounded like divisible by 20 or 100 is advisable.

Once you have been able to strictly define the region you be modeling you can collecting the GIS data you will need. Since you already know the pieces you will need, something for ground terrain, buildings, trees, water, and land use, you can start seeking out these pieces. Since most region will be centered on a particular city contacting that city's GIS office and asking what data they have will be most helpful. A city GIS office, if there is one, is going to be the best for having good quality and recent data so they will be using it the most often. With that said many cities can't afford or just don't have a GIS office, so in this case you can check if there is a statewide GIS office. Barring even that the USGS and the Census bureau have some data this is available, all of which is generally free. Links to those data source are listed later in this document.

Once you have been able to collect a sufficient set of data to being working with you can go ahead and do so, the specifics of this are in a further section in this document high-lighting the major steps in the processing timeline. The next sections of this document will describe the coordinate systems to be aware of as well as the principal data-types you will encounter with GIS data and how to handle them.

Coordinate Systems:
As mentioned previously, the final input data sets provided to produce Nomografs must be in the UTM Coordinate System. UTM, which stands for Universal Transverse Mercator, is a coordinate system, just like the probably more familiar geographic coordinate system which uses longitude and latitude to locate a point on the earth. UTM differs in that it is a Cartesian not geographic, which means it is broken into a gridded, equally sized set of cellular units, and not degree off sets from a reference angle that longitude and latitude would use.

As an example the longitude and latitude values for location of the Willis Tower in Chicago would be 41.8789, -87.6358, but when expressed in UTM it would be 447245 easting 4636526 northing, and UTM Zone 16T, northern hemisphere. The first thing to notice is that UTM contains a zone as part of

its expression, this zone referring to the designation of one of many zones all across the earth that UTM is defined by. Further information on these zones and more are presented later is this document. While long/lat and UTM are the typical coordinate systems there are in fact many others, such as the State Plane system, which is defined like UTM, except there is a separate set for every state in the United States. Various systems like this also exist for many other countries, defining their own zone and offset measurements. Because of this it is key to convert all of your data to UTM before working on it more deeply. Small aberrations can appear after cropping or resizing data and then switching coordinate systems later so it is strongly suggested to convert everything to UTM first so it can all be operated on congruently.

Typical File Types:
When collecting the GIS data it will come as two main kinds: raster and vector. The section below will briefly describe each of them. The point to note is that for use in ground terrain heights the raster data will almost always be the more useful and for the other types needed, buildings, trees, water, and land use, shape files will almost always be more helpful when they are available. Still, in a pinch, each kind can be usually be counted on to construct all the input datasets but the work to get there just may be more substantial.

Raster – Raster data can basically be thought of as an image, that is made up of a rectangular set of rows and columns where each pixel or cell is uniform in its dimensions and containing a value of some kind. For satellite imagery data that value would be a color, but for the purposes of Nomograf data you will want to find datasets that contain height values. Height values expressed as a raster will most likely be found as LIDAR data.

LIDAR stands for "Light Detection and Ranging" and is presented in a raster format, i.e. rows and columns with a different height-value at each point. LIDAR is most often collected by doing a flyover with special equipment which collects the highest point at each location, thereby collecting points not just along the ground but also treetops, rooftops, vehicle-tops, and otherwise. Because of this feature extraction is required to be done on LIDAR data in order to separate these components from one another to create a usable set of ground, buildings, trees, and otherwise. LIDAR is also typically found as two separate versions, known as passes, with the difference being the intensity of the laser that was used to measure the height, the first being less powerful than the second. This second more powerful pass penetrates further, passing through the leaves on trees or other smaller obstacles between the measure point and the ground. Having these two passes and noting their differences is what enables the feature extraction to be performed. LIDAR data typically comes encoded in GeoTiff format which is an image format with added meta-data for referencing where the data sits on Earth. Also, LIDAR can be encoded as LAS or ASCII which acts a point-cloud, described next.

Beyond LIDAR you will also find Digital Elevation Models (DEMs) for a lot of different regions, the primary difference here being that the resolution is usually less than what can be found in quality LIDAR and there will be only one set of the data, not the two passes found in LIDAR. This makes feature extraction not as possible with DEMs but often DEMS will already have removed most buildings and other large objects so it can be used more readily as a ground terrain height dataset assuming the resolution is of sufficient quality.

Vector – Vector data is different in that it represents shapes of things and is fixed only to those points which make up that shape, they are not gridded as the raster datasets are. This is useful because it means new shapes can be added or deleted readily, and more importantly these shapes can be categorized and visualized a variety of different ways because they will usually contain some attribute

information.

For instance if you had some vector data for all the buildings in a given city, each of the buildings could contain information specific to it, like it's name, address, tax zoning information, ownership name, number of occupants, etc. For our purposes in having something useful for building Nomograf datasets it could contain information relating to the height of the building. The reason it would be more useful to have this in vector and not raster is vector data can be calculated much rapidly and treating each object separately, unlike in a raster where it would just another set of pixels on a grid with no ability to delineate between the two. One of the most popular form of vector data is as Shapefiles.

Shapefiles are a format created by ESRI, the publishers of the GIS software ArcGIS. Shapefiles are a vector format, meaning it contains just point and line information, not image data as described for LIDAR or other raster formats. In many GIS databases this is the preferred way to render buildings, roads, specific locations, and even water. Shapfiles are broken in to separate pieces, with each piece being a specific shape type; polygon, polyline, or point.

As often as possible we recommend finding all the Nomografs datasets needed as Shapefiles, mostly since this is an industry standard and therefore a myriad of programs can make use of them. Also most of the hard work will be taken out of the processing, assuming height information is available. Even if there is no height information simply having the footprint areas of buildings can help you processing the data when using it to eliminate or highlight areas when processing from raster information. For non-buildings, like water and trees and roads and such, Shapefiles are very easy to use even when they lack additional information, as it safe to make greater assumptions about what and how to use them.

Processing Tools

In order to work with all of this GIS data you will obviously need some software tools. The tools we list below are the ones we are using or have used, but this is by no means a complete list. There are lots of GIS tools out there, some that do things better than others and some that don't. Also there are many that have features that others don't, but depending on what you need to do and what sort of conversion process you looking to do it may not matter.

ArcGIS - This is probably the most commonly used GIS software package in the industry. It given the option choose this tool, it supports a myriad of formats, what it can't use usually a way of converting to something it does is possible, and it has many plugins and script available for it that it enhance what it can do. The downside is it can be expensive, especially when using LIDAR Analyst or 3D Analyst or some of its more useful add-ons, but in most cases it will be a 90% solution.

QGIS – This is an open-source alternative to ArcGIS and is most notable in that it is free. It can also read a lot of different formats and also has a lot of free plugins and scripts to help it as well. It is probably as a good a solution as ArcGIS in many cases but it obviously won't have any full-on support available for it so you can't complain if it doesn't do everything right.

GDAL – Stands for Geospatial Data Abstraction Library is an open-source library that actually powers part many other GIS applications. By itself GDAL does nothing and must have people program to it in order to do anything, but many people already have so GDAL ends up being amazing for doing very small exacting tasks. GDAL runs command-line only and pre-compiled package of it's tools can be downloaded from its website. For bulk conversions or cropping and other such common tasks GDAL is often the best bet because it doesn't have the bulk the full GIS packages do.

ImageJ – In order to some of the raster processing you would like to do you will find ArcGIS and QGIS just aren't setup properly for it, particularly like adding values together from two sets of images. For this ImageJ works best and is open-source and free as well. It can be clunky to use sometime but it really will enable you to do some of those small tasks that are missing from the other tools used in the process.

Processing Timeline

Input Data Sets:

One of the first things to do after gathering your input data sets is to make sure can manage them properly. This entails tasks as mundane as renaming the files to something logical and relevant so you can keep track of what it is what and what changes have been made. For instance you would want to rename you initial set to something like “chicago_ground.tif” and “chicago_buildings.shp” for instance. If you later made many revisions you may have “chicago_ground_version2.tif” and “chicago_ground_version2_nobuildings.tif”. Obviously this can get quite confusing and those examples are just examples but you will want to come up with your own naming scheme that makes sense to you in order to be aware of how the process is going. Typically you are going to make a lot of mistakes, revisions, and temporary files over the course of the processing so having a logical way to track down the files is super important.

Geo-Referencing:

As mentioned earlier another first-step to take with your starting set of data is to geo-reference all of it to UTM. If you are using a GIS package like ArcGIS or QGIS this achieved most easily by importing all of the data you intend to use into the program. Then you need to set the global reference set inside of the program. What this means is while each of the pieces of data could have different geo-references like long/lat, and another already in UTM, and another in a State Plane, and another in some foreign countries custom reference set. When you set the global reference for the GIS program it will force all of the datasets to be rendered in this one reference set, however just because they are rendered/shown in this way it does not mean they are actually in this global format. You will need export each of the datasets into the global reference set type, which again should be UTM and using the appropriate zone for the region. This can be achieved usually by right clicking the data in the GIS application’s layer browser or also in one of the menus at the top, usually also for the layer. You will probably need to do this one by one for each of the datasets. Whenever asked you will want to use the WGS84 datum, sometimes also called the WGS84 reference ellipsoid.

One other key thing you will want to do as you export and rebuild each dataset into the UTM coordinate-space is to redefine the cell resolution units. Since datasets can come in at strange unit sizes like 5.434 feet per cell or something else strange like that you will want to make sure they all end up having common cell resolution units. In almost all cases you will want to export them into a cell resolution of 1 meter per cell, for both the X and the Y directions. If they are expressed as 1 meter per cell it is far easier to work with the UTM coordinate-space as it also is in 1 meter per cell units, so if you need to quick calculate something you can be safe all the units align to each other. In some cases exporting to 1 meter per cell will produce a file that is too large to actually use so you’ll want to export to 2 meters per cell, or 4 meters per cell, so on and so forth.

Feature Extraction:

The next step to take is to extract whatever features you are missing. In the best case you would receive a processed bare-earth LIDAR or DEM dataset for the ground, building shapefiles containing height attributes, and then shapefiles for trees, water, and roads. You may of course not receive it all this way or this cleanly. If that is the case you will need to extract the features you are missing.

To extract features you hopefully have 2 passes of LIDAR, as discussed briefly previously in this document. If you do then you can run tools on the data to figure out what is a building, what is a tree, what is road, etc. This can be a very time-consuming process and we will not cover that in detail here,

but suffice to say this is where tools like LIDAR Analyst or Feature Analyst can be of a great value. Also this is where possibly having imagery data is useful especially if it also comes with infrared data, since this cuts down or the search for trees and other vegetation.

If you are missing this 2 pass LIDAR then and you have no building information you will be in a very hard spot. Likely you will need to find stereographic pairs (2 sets of imagery of the same place from different angles) and then run programs that will be able extract buildings from this or find a GIS company who is capable of doing the work. In most processing runs for getting Nomograf datasets the time spent creating a building dataset is the most time consuming, particularly since the accuracy of the buildings is rather important, second only to probably the ground terrain heights.

If you have some building footprints with no height data and some good LIDAR data then you can do a process of "masking" the LIDAR and only keeping areas that fall within the building footprints, thus producing a height-field of the leftover raster points within the vector space.

Converting To Binary:

Once you have processed through the input datasets into something that is just the ground terrain, building, and trees you'll need to convert them to binary formats listed at the beginning of this document.

The first step is to get everything into a raster. While it was useful to work with much of the data as vector before you'll now need it all rasterized as this grid array format is how the Nomograf CFD codes need to read it in as. To convert the vector to raster the GIS application you are using will typically have a tool that can make this happen, if not GDAL provides this capability directly from Shapefile to GeoTIFF, the step for doing this is described in the Tips and Tricks section later in this document.

Once you have everything in raster, you can use the GDAL command to perform the last bit of steps, namely: 1) multiplying all the floating point height values by 20 and then truncating off the decimal places, and then 2) saving the data as two-byte integers that are unsigned fixed. ImageJ also provides a capability of doing this as well.

Tips and Tricks

This section is presented ad-hoc, with suggested command-line commands that can be performed using the GDAL toolset. If you do not have this toolset you can download it from the location mentioned in the links and references section.

To join several shapefiles and keep only ELEVATION:

```
ogr2ogr -select ELEVATION low_res.shp low_res1.shp
ogr2ogr -update -append -select ELEVATION low_res.shp low_res4.shp -nln
low_res
ogr2ogr -update -append -select ELEVATION low_res.shp low_res3.shp -nln
low_res
ogr2ogr -update -append -select ELEVATION low_res.shp low_res2.shp -nln
low_res
```

To sort a shapefile in ascending order:

```
ogr2ogr -sql "SELECT * FROM low_res ORDER BY ELEVATION ASC"
sorted_asc_low_res.shp low_res.shp
```

To select all features that have a specific value for an attribute:

```
ogr2ogr -sql "SELECT * FROM tl_2009_06037_edges WHERE MTFCC = 'S1200'"
S1200.shp tl_2009_06037_edges.shp
```

To delete polygons with small height:

```
ogr2ogr -sql "SELECT * FROM sorted WHERE (HEIGHT_ROO >= 4.00)" sorted2.shp
sorted.shp
```

To reproject a shapefile which contains a coordinate system:

```
ogr2ogr -t_srs EPSG:32611 S1730_utm.shp S1730.shp
```

To convert KML to shapefile

```
ogr2ogr -nlt POLYGON -t_srs EPSG:32619 delete_trees.shp delete_trees.kml
```

To get info about a shapefile:

```
ogrinfo -summary S1740.shp S1740
```

To get info about a geotiff:

```
gdalinfo low_res_sorted_asc.tif
```

To rasterize a shapefile using the feature id:

```
gdal_rasterize -a FID -sql "select FID, * from sorted_asc_low_res"
sorted_asc_low_res.shp low_res_numbers.tif
```

To rasterize a shapefile and assign it a fixed value:

```
gdal_rasterize -l S1730_utm -burn 30 S1730_utm.shp roads.tif
```

To rasterize a shapefile with a fixed value for a specified extent into a new raw binary file:

```
gdal_rasterize -l All_Boston_Water_UTM_CUT -init 0 -burn 1 -a_nodata 0 -
a_srs EPSG:32619 -te 323100.0 4685400.0 337100.0 4695400.0 -tr 1.0 1.0 -ot
INT16 -of ENVI All_Boston_Water_UTM_CUT.shp water.bin
```

To rasterize multiply by twenty and save as a binary file:

```
1. gdal_rasterize -l sorted2 -a HEIGHT_ROO -init 0 -a_nodata 0 -a_srs
EPSG:32618 -tr 1.0 1.0 sorted2.shp bldgs1.tif
2. gdal_translate -scale 0 1 0 20 -ot INT16 -of ENVI bldgs1.tif bldgs.bin
```

To overlay two geotiffs (using 0 as the nodata value):

```
gdalwarp -dstnodata 0 LowResBuildings.tif HighResBuildings.tif
Buildings.tif
```

To merge a bunch of files in directories into one tif with a selected region and coordinate change:

```
gdalwarp -t_srs EPSG:32611 -te 380400 3762000 390400 3772000 -tr 1 1 -r
bilinear -ot Float32 -multi 6471* 6477*  terrain.tif
```

To save out the raw binary raster:

```
gdal_translate -of ENVI water_masked_buildings.vrt test.bin
```

To add a projection, override extents, and cut a selection:

```
gdal_translate -a_srs EPSG:32632 -a_ullr 563010 5935436 567010 5931436 -
a_nodata 0 -srcwin 441 441 1600 1600 roads_with_names.png
roads_with_names_trimmed.tif
```

To reproject for Google Earth:

```
gdalwarp -t_srs EPSG:4326 -co "COMPRESS=DEFLATE" raw_tree_heights.vrt
raw_tree_heights_for_ge.tif
```

Links and References:

Coordinate Related:

Universal Transverse Mercator coordinate system
http://en.wikipedia.org/wiki/Universal_Transverse_Mercator_coordinate_system

Cartesian coordinate system
http://en.wikipedia.org/wiki/Cartesian_coordinate_system

World Geodetic System
http://en.wikipedia.org/wiki/World_Geodetic_System

Reference ellipsoid
http://en.wikipedia.org/wiki/Reference_ellipsoid

State Plane Coordinate System
http://en.wikipedia.org/wiki/State_plane

UTM Grid Zones of the World
http://www.dmap.co.uk/utmworld.htm

Software Related: ArcGIS
http://www.esri.com/software/arcgis

QGIS
http://www.qgis.org/

ImageJ
http://rsbweb.nih.gov/ij/

GDAL
http://www.gdal.org/

File Format Related: Shapefile
http://en.wikipedia.org/wiki/Shapefile

GeoTIFF
http://en.wikipedia.org/wiki/Geotiff

Digital Elevation Model (DEM)
http://en.wikipedia.org/wiki/Digital_elevation_model

LIDAR
http://en.wikipedia.org/wiki/Lidar